PL-TR-95-2132
Environmental Research Papers, No. 1180

LONG-RANGE LATTICE-GAS SIMULATION ON THE CAM-8 PROTOTYPE

Jeffrey Yepez

21 September 1995

PHILLIPS LABORATORY
Directorate of Geophysics
AIR FORCE MATERIEL COMMAND
HANSCOM AIR FORCE BASE, MA 01731-3010

"This technical report has been reviewed and is approved for publication."

Donald D Grantham
DONALD D. GRANTHAM
Chief, Atmospheric Structure Branch
Atmospheric Sciences Division

Robert A. McClatchey
ROBERT A. McCLATCHEY
Director, Atmospheric Sciences Division

This report has been reviewed by the ESC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify PL/IM, 29 Randolph Street, Hanscom AFB, MA 01731-3010. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** 21 September 1995	**3. REPORT TYPE AND DATES COVERED** Final Report

4. TITLE AND SUBTITLE Long-Range Lattice-Gas Simulation on the CAM-8 Prototype	**5. FUNDING NUMBERS** PE 61102F PR 2304 TA CP WU 10
6. AUTHOR(S) Yepez, Jeffrey	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Phillips Laboratory (GPAA) 29 Randolph Rd Hanscom AFB, MA 01731-3010	**8. PERFORMING ORGANIZATION REPORT NUMBER** PL-TR-95-2132 ERP, No. 1180
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research Mathematical and Computational Sciences	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(Maximum 200 words)*

A novel algorithmic method presented for simulating complex fluids, for instance multiphase single component fluids and molecular systems. The algorithm falls under a class of single-instruction multiple-data computation known as lattice-gases, and therefore inherits exact computability on a discrete spacetime lattice. Our contribution is the use of nonlocal interactions that allow us to model a richer set of physical dynamics, such as crystallization processes, yet to do so in a way that remains locally computed. A simple computational scheme is employed that allows all the dynamics to be computed in parallel with two additional bits of local site data, for outgoing and incoming messengers regardless of the number of long-range neighbors. The computational scheme is an efficient decomposition of a lattice-gas with many neighbors. It is conceptually similar to the idea of virtual intermediate particle momentum exchanges that is well known in particle physics. All 2-body interactions along a particular direction define a spatial partition that is updated in parallel. Random permutation through the partitions is sufficient to recover the necessary isotropy as long as enough momentum exchange directions are used. The algorithm is implemented on the CAM-8 prototype.

14. SUBJECT TERMS Lattice-gas　　Complex fluids Parallel computing　　Liquid-gas transitions Multiphase fluids　　Cellular automata	**15. NUMBER OF PAGES** 50 **16. PRICE CODE**

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** SAR

Contents

Illustrations

Tables

1 Introduction

To illustrate the how one may implement a lattice-gas with long-range interactions, let us consider for simplicity a two-dimensional example with a system having only one interaction range and consider only an attractive interaction. The more general case of multiple interaction ranges with both repulsive and attractive interactions and in higher dimensions will follow directly. Specifically, we discuss our new molecular dynamics lattice-gas algorithm that uses eight interaction ranges and both repulsive and attractive interactions to approximate a Lenard-Jones intermolecular potential.

A long-range lattice-gas has been implemented on the MIT cellular automata machine prototype, the CAM-8. Consequently, first the CAM-8 architecture is briefly described. Next a brief description of what a lattice-gas automaton is and explain why it is an exactly computable representation of a dynamical system is given. One of the principal requirements for a lattice-gas with microscopic finite-point group symmetry to give rise to macroscopic continuous rotational symmetry is that the underlying lattice must be *isotropic*. Therefore I describe what it means for a lattice to be isotropic. Working in two dimensions is much easier than working in three, both for implementing computer models and for describing them. For this reason I present the long-range lattice-gas algorithm in two dimensions on the triangular lattice.

When introduced to the triangular lattice-gas model for the first time, one inevitably asks the following question: Why does discrete dynamics fail

Received for publication 15 September 1995

to reproduce the correct continuum hydrodynamic limit when implemented on a square lattice? One finds that four momentum states are insufficient by noting that the derivation of the Navier-Stokes equation relies on the expansion of the momentum flux density tensor in terms of the isotropic tensor E^4. In turn the E^4 tensor could be expanded in products of two dimensional Kronecker deltas, given below in Eq. (10). For the square lattice case, the lattice vectors are orthogonal and E^4 cannot be decomposed into two-dimensional Kronecker deltas. Instead

$$E^4_{ijkl}|_{B=4} = 2\delta_{ijkl}$$

where δ_{ijkl} is a four dimensional Kronecker delta, illustrating the lack of isotropy of the momentum flux density on a square lattice-gas. Since five nearest neighbors are not space filling, the next possible choice is six or the triangular lattice. The simplest discrete dynamics in two dimensions is known as a hexagonal lattice-gas or an FHP lattice-gas, after its originators Uriel Frisch, Brosl Hasslacher, and Yves Pomeau [1]. Since the long-range lattice-gas still retains local collisions, the simple FHP model is presented here for completeness. Next we examine the long-range 2-body interaction, restricting ourselves to central-body attractive interactions, for the sake of simplicity. Two different bound states, the bounce-back orbit and the clockwise orbit are discussed.

When implementing lattice-gas algorithms it is often useful try to find economical ways of expressing the collisions or interactions, to reduce the size of a look-up table or reduce the depth of the logical representation of the algorithm. To this end I briefly discuss some symmetries inherent in long-range interactions, in particular I introduce parity and conjugation symmetries. Finally, I discuss the implementation of a multi-long-range lattice-gas. Remarkably, this methodology allows us to model solid-state dynamics and as such offers an alternative to the usual method of computational molecular dynamics.

2 The Cellular Automata Machine CAM-8

The cellular automata machine CAM-8 architecture devised by Norman Margolus of the MIT Laboratory for Computer Science [2, 3] is the latest in a line of cellular automata machines developed by the Information Mechanics Group at MIT [4, 5, 6]. It is optimized for performing lattice-gas simulations. The CAM-8 architecture itself is a simple abstraction of lattice gas dynamics. Lattice gas data streaming and collisions are directly implemented in the architecture. The communication network is a cartesian three dimensional mesh. Crystallographic lattice geometries can be directly embedded into the CAM-8. Each site of the lattice has a certain number of bits (a multiple of 16) which we refer to as a "cell". Each bit of the cell, or equivalently each bit plane of the lattice, can be translated through the lattice in any arbitrary direction. The translation vectors for the bit planes are termed "kicks". The specification of the x,y, and z components of the kicks for each bit plane (or

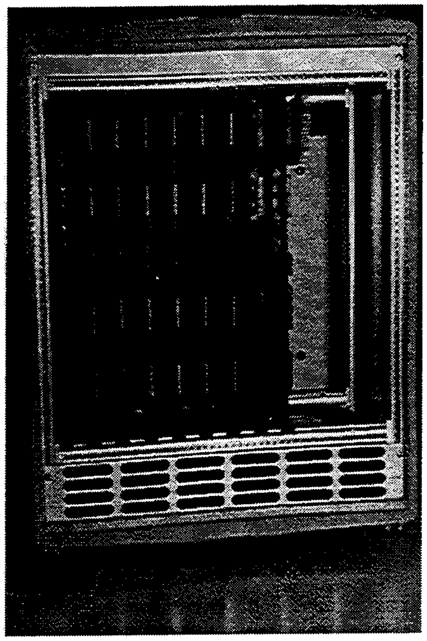

Figure 1: MIT Laboratory for Computer Science Cellular Automata Machine CAM-8. This 8 module prototype can evolve a D-dimensional cellular space with 32 million sites where each site has 16 bits of data with a site update rate of 200 million per second.

hyperplane) exactly defines the lattice. The kicks can be changed during the simulation. Thus, the data movement in the CAM-8 is general. Once the kicks are specified, the coding of the lattice-gas streaming is completed. In effect, the kicks determine all the global permutations of the data.

Every configuration of the local data within a cell is an element of an *equivalence class* having a particular value of the conserved quantities of the dynamics. Local permutations of data in an equivalence class occur within the cells. These local permutations are the computational metaphor for physical conservative collisions between particles.[1] All local permutations are implemented in look-up tables. That is, all possible physical events with a certain input configuration and a certain output configuration are precomputed and stored in SRAM, for fast table look-up. The width of the CAM-8 look-up tables are limited to 16-bits, or 64K entries. This is a reasonable width, satisfying the opposing considerations of model complexity versus memory size limitations for the SRAM. Site permutations of data wider than 16-bits must be implemented in several successive table look-up passes. Since the look-up tables are double buffered, a scan of the space can be performed while a new look-up table is loaded for the next scan.

Figure 2 is a schematic diagram of a CAM-8 system. On the left is a single hardware module—the elementary "chunk" of the architecture. On the right is an indefinitely extendable array of modules (drawn for convenience as two-dimensional, the array is normally three-dimensional). A uniform spatial calculation is divided up evenly among these modules, with each module able to simulate a volume containing millions of fine-grained spatial sites in a sequential fashion. In the diagram, the solid lines between modules indicate a local *mesh* interconnection. These wires are used for spatial data movements. There is also a tree network (not shown) connecting all modules to the front-end host, a SPARCstation with a custom SBus interface card.

[1] The CAM-8 is not limited to performing only local permutations; it can do general mappings. However, since we are interested only in particle conserving reversible dynamics, local permutations are sufficient.

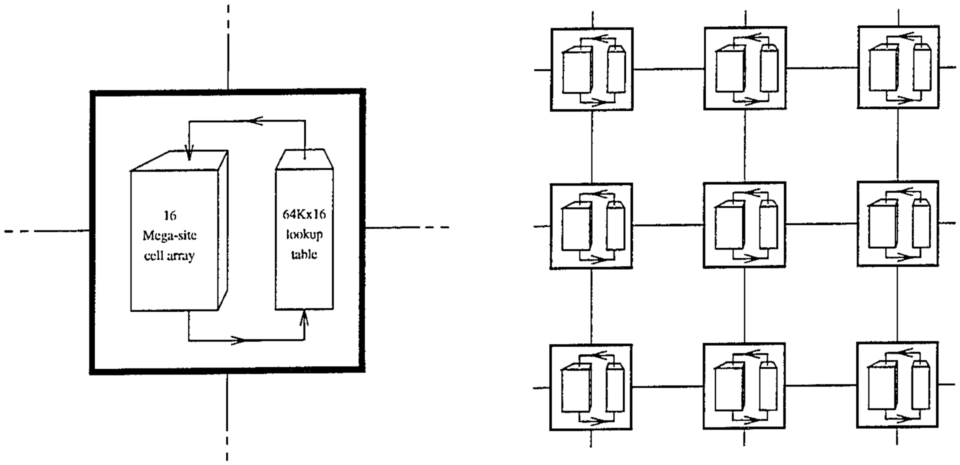

Figure 2: CAM-8 System Diagram. (a) A single processing node, with DRAM site data flowing through a SRAM lookup table and back into DRAM. (b) Spatial array of CAM-8 nodes, with nearest-neighbor (mesh) interconnect (1 wire/bit-slice in each direction).

This SPARCstation controls the CAM-8. It downloads a bit-mapped pattern as the initial condition for the simulations. It also sends a "step-list" to the CAM-8 to specify the sequence of kicks and scans that evolve the lattice-gas in time. One can view the lattice-gas simulation in real-time since a custom video module captures site data for display on a VGA monitor, a useful feature for lattice-gas algorithm development, test and evaluation. The CAM-8 has built-in 25-bit event counters allowing real-time measurements without slowing the lattice-gas evolution. This feature is used to do real-time coarse-grain block averaging of the lattice-gas number variables and to compute the components of the momentum vectors for each block. The amount of coarse-grained data is sufficiently small to be transferred back to the front-end host for graphical display as an evolving flow field within an X-window.

3 Lattice-Gas Automaton: An Exactly Computable Dynamical System

A Boolean formulation of an exactly computable dynamical system, known as a lattice-gas, may be stated in a way that is consistent with the Boltzmann equation for kinetic transport. In essence the lattice-gas dynamics are a simplified form of molecular transport as we restrict ourselves to a cellular phase space. The macroscopic equations, in particular the continuity equation and the Navier-Stokes equation, are obtained by coarse-graining over a discrete microdynamical transport equation for a number of Boolean variables. The scheme employs the finite-point group symmetry of a crystallographic spatial lattice. It is somewhat inevitable that to obtain an exactly computable representation of fluid dynamics one must perform a statistical treatment over discrete number variables.

Before the basic lattice-gas microdynamical transport equation is introduced, some notational conventions are needed. Consider a spatial lattice with N total sites. The fundamental unit of length is the size of a lattice cell, l, and the fundamental unit of time, τ, is the time it takes for a speed-one particle to go from one lattice site to a nearest neighboring site. Particles, with unit mass m, propagate on the lattice. The unit lattice propagation speed is denoted by $c = \frac{l}{\tau}$. Particles occupy this discrete space and can have only a finite number B of possible momenta. The lattice vectors are denoted by e_{ai} where $a = 1, 2, \ldots, B$. For example, for a single-speed gas on a triangular lattice, $a = 1, 2, \ldots, 6$. A particle's state is completely specified at some time, t, by specifying its position on the lattice, x_i, and its momentum, $p_i = mce_{ai}$, at that position. The particles obey Pauli exclusion since only one particle can occupy a single momentum state at a time. The total number of configurations per site is 2^B. The total number of possible single particle momentum states available in the system is $N_{\text{total}} = BN$. With P particles in the system, denote the filling fraction by $d = \frac{P}{N_{\text{total}}}$.

The number variable, denoted by $n_a(\mathbf{x}, t)$, takes the value of one if a particle exists at site $\mathbf{x}$ at time t in momentum state $mc\ \hat{\mathbf{e}}_{\mathbf{a}}$, and takes the value of zero otherwise. The evolution of the lattice-gas can then be written in terms of n_a as a two-part process: a collision part and a streaming part. The collision part reorders the particles locally at each site.

$$n'_a(\mathbf{x}, t) = n_a(\mathbf{x}, t) + \Omega_a(\vec{n}(\mathbf{x}, t)), \tag{1}$$

where Ω_a represents the collision operator and in general depends on all the particles, $\vec{n}$ at the site. So as a shorthand suppress the index on the occupation variable when it is an argument of $\Omega_a[\vec{n}(\mathbf{x}, t)]$ to represent this general dependence. In the streaming part of the evolution the particle at position $\mathbf{x}$ "hops" to its neighboring site at $\mathbf{x} + l\hat{\mathbf{e}}_{\mathbf{a}}$ and then time is incremented by

τ

$$n'_a(\mathbf{x} + l\hat{\mathbf{e}}_\mathbf{a}, t + \tau) = n_a(\mathbf{x}, t) + \Omega_a[\vec{n}(\mathbf{x}, t)]. \tag{2}$$

Equation (2) is the lattice-gas microdynamical transport equation of motion.

The collision operator can only permute the particles locally on the site since we wish the local particle number to be conserved before and after the collision. That is,

$$n(\mathbf{x}, t) = \sum_a n'_a(\mathbf{x}, t) = \sum_a n_a(\mathbf{x}, t). \tag{3}$$

Equation (3) defines the local number density. Summing Eq. (2) over lattice directions then implies the constraint on the collision operator,

$$\sum_a \Omega_a = 0. \tag{4}$$

We may define the local momentum as

$$p_i(\mathbf{x}, t) = mc \sum_a e_{ai} n'_a(\mathbf{x}, t) = mc \sum_a e_{ai} n_a(\mathbf{x}, t), \tag{5}$$

which of course must also be conserved before and after a collision. Again, this imposes a constraint on the collision operator.

$$\sum_a e_a \Omega_a = 0. \tag{6}$$

As a matter of notation it should be understood that whenever a directionally dependent quantity is written, its subscripted index is taken modulo B. Using the number variable, for example, it is understood that

$$n_{a+b} = n_{\mathrm{mod}_B (a+b)}. \tag{7}$$

As a shorthand, a negative index will represent the antiparallel direction, so since $\hat{\mathbf{e}}_{a+\frac{B}{2}} = -\hat{\mathbf{e}}_a$ we may write

$$n_{-a} = n_{a+\frac{B}{2}}. \tag{8}$$

4 Isotropic Lattice Tensors

We construct an n^{th} rank tensor composed of a product of lattice vectors [7]

$$E^{(n)} = E_{i_1 \ldots i_n} = \sum_a (e_a)_{i_1} \cdots (e_a)_{i_n}, \tag{9}$$

where $a = 1, \ldots, B$. All odd rank E vanish. We wish to express $E^{(2n)}$ in terms of Kronecker deltas, $\delta_{ij} = 1$ for $i = j$ and zero otherwise. We can turn this problem of expressing the E-tensors in terms of products of Kronecker deltas into a problem of combinatoric counting. We use the following tensors:

$$\begin{aligned} \Delta^2_{ij} &= \delta_{ij} && (10) \\ \Delta^4_{ijkl} &= \delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{kj} && (11) \end{aligned}$$

and so forth. Then we know that if E is isotropic it must be proportional to Δ, therefore

$$E^{(2n)} \propto \Delta^{(2n)} \tag{12}$$

and that the constant of proportionality may be obtained by counting the number of ways we could write a term comprising a product of n Knonecker deltas. Consider for example the case $n = 2$. Since the Knonecker delta is symmetric in its indices, the following four products are identical: $\delta_{ij}\delta_{kl} =$

$\delta_{ij}\delta_{lk} = \delta_{ji}\delta_{kl} = \delta_{ji}\delta_{lk}$. The degeneracy is 2^2. Furthermore, the order of the Kronecker deltas also doesn't matter since they commute; that is, $\delta_{ij}\delta_{kl} = \delta_{kl}\delta_{ij}$. The degeneracy is 2!. For the case where n is arbitrary, there are 2^n identical ways of writing the product of n Kronecker deltas. For each choice of indices, there are an additional $n!$ number of ways of ordering the products. Therefore, the total number of degeneracies equals $2^n n! = (2n)!!$. The total number of permutations for $2n$ indices equals $(2n)!$. So from this counting procedure we know that $\Delta^{(2n)}$ consists of a sum of $\frac{(2n!)}{(2n)!!} = (2n-1)!!$ terms.

The following relations will be very useful throughout later developments

$$\begin{aligned} E^1 &= 0 && (13) \\ E^2 &= \frac{B}{D}\delta_{ij} && (14) \\ E^3 &= 0 && (15) \\ E^4 &= \frac{B}{D(D+2)}\left(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{kj}\right) && (16) \end{aligned}$$

In general, the lattice tensors are

$$\begin{aligned} E^{2n+1} &= 0 && (17) \\ E^{2n} &= \frac{B}{D(D+2)\cdots(D+2n-2)}\Delta^{2n} && (18) \end{aligned}$$

5 Triangular Lattice

In a triangular lattice there are six vectors, enumerated in this section by the convention

$$\hat{\mathbf{e}}_{\mathrm{a}} = \left(\cos\frac{\pi a}{3}, -\sin\frac{\pi a}{3}\right), \tag{19}$$

where $a = 1, 2, \ldots, 6$. The spatial coordinates of the lattice sites may be

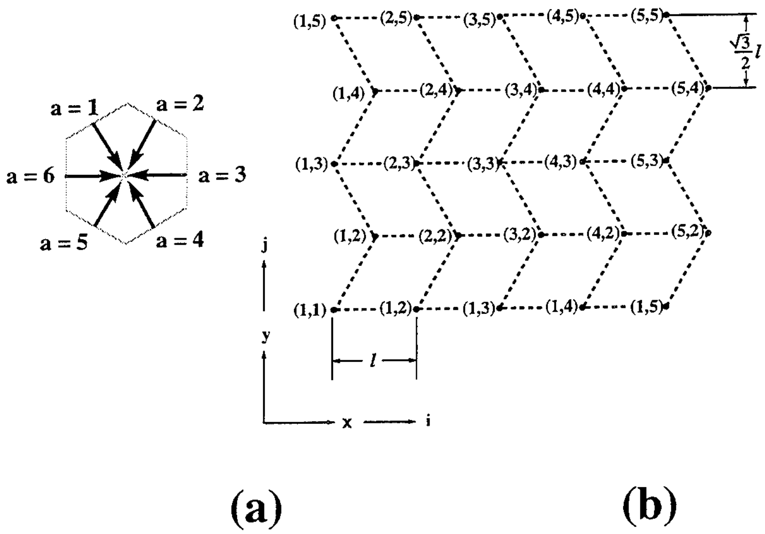

Figure 3: Triangular Lattice Convention: (a) Lattice vector label convention; (b) Triangular lattice convention with lattice directions $a = 3$ up and $a = 6$ down. Coordinates above the lattice nodes are (i, j) memory array indices.

expressed as follows

$$\mathbf{x}_{ij} = \left(i - \frac{1}{2}(j \bmod 2), \frac{\sqrt{3}}{2}j\right) \tag{20}$$

where i and j are rectilinear indices that specify the data memory array location used to store the lattice-gas site data.

Let $s = (j \bmod 2)(r \bmod 2)$. Given a particle at site (i, j), it may be shifted to a site r lattice units away to a remote site (i', j') by the mapping

$$
\begin{aligned}
(i', j')_1 &= \left(i + \frac{r+1}{2} - s, j - r\right) && (21)\\
(i', j')_2 &= \left(i - \frac{r}{2} - s, j - r\right) && (22)\\
(i', j')_3 &= (i - r, j) && (23)\\
(i', j')_4 &= \left(i - \frac{r}{2} - s, j + r\right) && (24)\\
(i', j')_5 &= \left(i + \frac{r+1}{2} - s, j + r\right) && (25)\\
(i', j')_6 &= (i + r, j) && (26)
\end{aligned}
$$

where $(i', j')_a$ denotes the shifted site, that is, $(i, j) \rightarrow (i', j')$ with a shift along vector $\vec{\mathbf{r}} = r\hat{\mathbf{e}}_a$ and where division by 2 is considered integer division.

6 Local Collision Rules

In two dimensions we may use a triangular lattice, with six bits per site encoding the occupation numbers of the six possible momentum states. Let n_a be the input bits and n'_a be the output bits of a local collision. A general collision operator is constructed as follows

$$\Omega_a = \sum_{\{\zeta_i\}} \alpha Q_a(\{\zeta_i\}), \qquad (27)$$

where $\{\zeta_i\}$ is a set of occupied particle states, $\alpha = \pm 1$ is a scalar coefficient, and where each term in the sum is written in factorized form as

$$Q_a(i_1, \ldots, i_k) = \frac{n_{a+i_1}}{1 - n_{a+i_1}} \cdots \frac{n_{a+i_k}}{1 - n_{a+i_k}} \prod_{j=1}^{B} (1 - n_{a+j}). \qquad (28)$$

Table 1: Simple Right-Handed Collision Table

n_0	n_1	n_2	n_3	n_4	n_5	n'_0	n'_1	n'_2	n'_3	n'_4	n'_5
1	0	0	1	0	0	0	0	1	0	0	1
0	1	0	0	1	0	1	0	0	1	0	0
0	0	1	0	0	1	0	1	0	0	1	0
1	0	1	0	1	0	0	1	0	1	0	1
0	1	0	1	0	1	1	0	1	0	1	0

Then the FHP collision operator is the following:

$$\Omega_a^{\text{FHP}} = \frac{1}{2}Q_a(1,4) + \frac{1}{2}Q_a(2,5) - Q_a(0,3) + Q_a(1,3,5) - Q_a(0,2,4) \qquad (29)$$

or for $a = 0$ this is explicitly

$$\begin{aligned} \Omega_0 \quad = \quad & \frac{1}{2}n_1 n_4(1-n_0)(1-n_2)(1-n_3)(1-n_5) + \\ & \frac{1}{2}n_2 n_5(1-n_0)(1-n_1)(1-n_3)(1-n_4) - \\ & n_0 n_3(1-n_1)(1-n_2)(1-n_4)(1-n_5) + \\ & n_1 n_3 n_5(1-n_0)(1-n_2)(1-n_4) - \\ & n_0 n_2 n_4(1-n_1)(1-n_3)(1-n_5). \end{aligned}$$

Note that it is sufficient to give only Ω_0 since the other components of the collision operator can be obtained simply by incrementing the indices of Ω_0 owing to the six-fold symmetry of the collisions. The factors of one-half in Eq. (29) are transition probabilities for the 2-body collisions, indicating a coin toss is performed to choose between even or odd chirality.

Table 2: Simple Left-Handed Collision Table

n_0	n_1	n_2	n_3	n_4	n_5	n'_0	n'_1	n'_2	n'_3	n'_4	n'_5
1	0	0	1	0	0	0	1	0	0	1	0
0	1	0	0	1	0	0	0	1	0	0	1
0	0	1	0	0	1	1	0	0	1	0	0
1	0	1	0	1	0	0	1	0	1	0	1
0	1	0	1	0	1	1	0	1	0	1	0

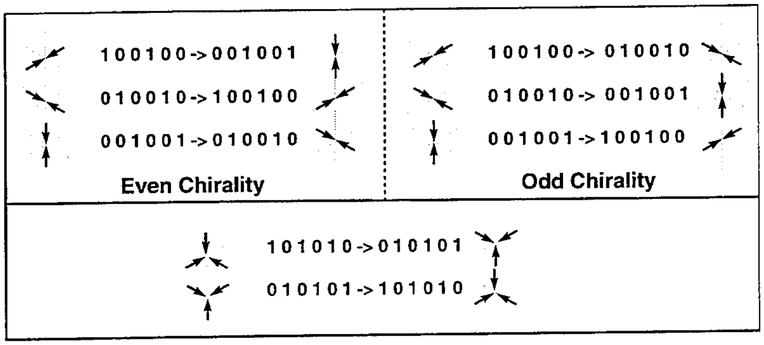

Figure 4: FHP Collision Rules. Enumeration of FHP 2-body collisions, even and odd chirality, and 3-body collisions.

The possible two-body and three-body collisions represented by Eq. (29) are illustrated in Figure 4. For two-dimensional flow, there are four invariants, the mass, two components of the momentum, and the energy. With only the 2-body collision in Figure 4, there is an additional invariant: the difference in the particle number along each of the three lattice directions. The 3-body collisions in Figure 4 were include in the FHP-model to remove this spurious invariant. Consequently, the collisions enumerated in Figure 4 are the minimally sufficient set to produce hydrodynamic behavior in the continuum limit.

7 Long-Range 2-Body Interactions

An interparticle potential, $V(\mathbf{x} - \mathbf{x}')$, acts on particles spatially separated by a fixed distance, $\mathbf{x} - \mathbf{x}' = \mathbf{r}$. An effective interparticle force is caused by a non-local exchange of momentum. Momentum conservation is violated locally, yet it is exactly conserved in the global dynamics.

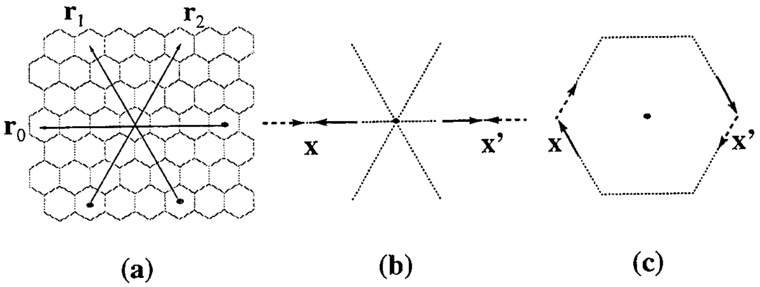

Figure 5: Bounce-Back and Clockwise Bound States. Simple bound-state orbits due to a long-range attractive interaction where the dotted path indicates the particle's closed trajectory: (a) partition directions; (b) bounce-back orbit with $|\Delta p| = 2$ and zero angular momentum; and (c) clockwise orbit with $|\Delta p| = 1$ and one unit of angular momentum. Head of the dashed arrows indicates particles entering the sites of partition $\mathbf{r}_0$ at time t. Tail of the black arrows indicates particles leaving those sites at time $t + \tau$. The counter-clockwise orbit is not shown.

For an attractive interaction, there exists bound states in which two particles orbit one another. Since the particle dynamics are constrained by a

Table 3: Lattice Vector Components

a	x-component	y-component
0	-1	0
1	$-\frac{1}{2}$	$\frac{\sqrt{3}}{2}$
2	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$
3	1	0
4	$\frac{1}{2}$	$-\frac{\sqrt{3}}{2}$
5	$-\frac{1}{2}$	$-\frac{\sqrt{3}}{2}$

crystallographic lattice we expect polygonal orbits. In Figure 5 we have depicted two such orbits for a hexagonal lattice-gas. The range of the interaction is r. Two-body single range attractive interactions are depicted in Figures 5b and 5c, the bounce-back and clockwise orbits respectively. Momentum exchanges occur along the principal directions. The interaction potential is not spherically symmetric, but has an angular anisotropy. In general, it acts only on a finite number of points on a shell of radius $\frac{r}{2}$. The number of lattice partitions necessary per site is half the lattice coordination number, since two particles lie on a line. Though microscopically the potential is anisotropic, in the continuum limit obtained after coarse-grain averaging, numerical simulation indicates isotropy is recovered.

8 A Simple Example: Bounce-Back Orbit

A long-range lattice-gas of the type we are considering still possesses the usual local dynamics of a hydrodynamic lattice-gas. To extend the local lattice-gas update rules to include long-range interactions, we use two additional bits of local site data. This will allow us to implement a long-range interaction using strictly local updating and therefore the algorithm remains "embarrassingly" parallel just as a usual local lattice-gas. The two additional bits will denote the occupation numbers of messenger particles, or "photons". The idea of using messenger particles was introduced by Appert *et al.*[8]. We have two types of messenger states, to represent incoming and outgoing conditions, and we denote the messengers as z_l and z_r.

For the simplest long-range lattice-gas model, we therefore use eight bits of local site data. Since long-range interactions occur between remote spatial sites, say $\vec{x}$ and $\vec{x}'$, the messenger particles will travel either parallel or antiparallel to the vector $\vec{r} = \vec{x} - \vec{x}'$. All pairs of sites throughout the entire space that are separated by the vector $\vec{r}$ can therefore all be updated in parallel. We refer an update step of all pairs of 2-body interactions along direction $\vec{r}$ as a *partition*, denoted by Γ_r. All possible two-body interaction pairs are then computed by performing all possible partitions of the space. For this reason, it requires many scans for the space to perform a single long-range interaction step.

In our two-dimensional example using a triangular lattice, there are three possible partitions. The number of partitions is never smaller than half the lattice coordination number. In the two-dimensional case, the simplest long-range lattice-gas algorithm, though perhaps not the most efficient algorithm, is to use three *sequential* scans of the space, each scan performing the updating necessary for a single partition. See Figure 5a. Often, depending on the complexity of the long-range interactions and the dimensionality of the lattice, it is possible to perform *simultaneous* updating of multiple partitions. This of course is more desirable, yet it causes more complexity. Furthermore, this updating requires an extra pair of messenger particles for each partition to be simultaneously updated. For simplicity, we will not deal with this case here, however our implementation on the CAM-8 does use simultaneous partition updating—repulsive and attractive partitions are performed in parallel using four messenger bits.

Let us consider a simple example of the long-range lattice-gas algorithm, the minimal model of Appert. Here we consider only *bounce-back* attractive interactions. Suppose there is a single particle at site $\vec{x} = 0$ and there is also a single particle at site $\vec{x}' = r\hat{\mathbf{i}}$; that is, $n_0(\vec{x}) = 1$, $n_3(\vec{x}) = 0$, $n_0(\vec{x}') = 0$ and $n_3(\vec{x}') = 1$ with all other bits at $\vec{x}$ and $\vec{x}'$ being zero. See Figure 5b for a diagram of this situation. Here we are using the bit convention shown in table 3. Then the two particles are separated by a distance r and are moving away from each other. The attractive long-range interaction will effectively flip their respective directions making $n_0(\vec{x}) = 0$, $n_3(\vec{x}) = 1$, $n_0(\vec{x}') = 1$ and $n_3(\vec{x}') = 0$ so that the two particles will now be moving toward each other. There is a local momentum change of $2mc\hat{\mathbf{i}}$ at $\vec{x}'$ and an opposite momentum change of $-2mc\hat{\mathbf{i}}$ at $\vec{x}$. Locally momentum is not conserved, but nonlocally it is.

The first step of the long-range interaction sequence is to choose a partition, say Γ_r, and then to emit messenger particles along the partition axis. The basic local rule for this first step is the following: a photon is emitted at a site if there exists a particle at that site that can partake in a long-range interaction. Another way of expressing this rule is: *send only if you can receive.* Obviously, for a particle to partake in an interaction there must be both a particle and a hole at that site. The factorized probability of having such a situation is just $d(1-d)$. So to continue with our example, for a photon to be emitted at some site $\vec{x}$ parallel or antiparallel to a partition direction $\hat{\mathbf{i}}$, we use the following rule

$$\begin{aligned} z_r(\vec{x}) &= n_0(\vec{x})(1-n_3(\vec{x})) & (30) \\ z_l(\vec{x}) &= n_3(\vec{x})(1-n_0(\vec{x})). & (31) \end{aligned}$$

Note that according to this local rule, only one photon can be created at a site, and consequently we eliminate the possibility of a long-range interaction, say of range $2r$, mediated through a doubly occupied site. For two sites separated by the interaction distance r, the important consequence of the emission step is that if both sites send photons, both will necessarily receive them, which strictly enforces nonlocal momentum conservation. *Give and ye shall receive* (provided yours is received). Letting $z_a \equiv z_r$ and $z_{-a} \equiv z_l$, in general we can write the emission step of the minimal interaction as

$$z_a(\vec{x}) = n_{-a}(\vec{x})(1-n_a(\vec{x})), \tag{32}$$

where $a = 0, 1, 2$ covers all the partitions.

After the emission step, a long-range kick of the messenger bits follows. In the simple example, all photons z_l are kicked along $-r\hat{\mathbf{i}}$ and all photons z_r are kicked along $r\hat{\mathbf{i}}$. In general for the long-range kick we have

$$z'_a(\vec{x} + r\hat{e}_a) = z_a(\vec{x}). \tag{33}$$

Finally, we have the absorption step of the long-range interaction sequence. Here the local particle momentum state is updated as the particles flip their directions in our example

$$n'_3(\vec{x}) = n_3(\vec{x}) + z'_l(\vec{x})n_0(\vec{x})(1 - n_3(\vec{x})) - z'_r(\vec{x})n_3(\vec{x})(1 - n_0(\vec{x})) \tag{34}$$

$$n'_0(\vec{x}) = n_0(\vec{x}) + z'_r(\vec{x})n_3(\vec{x})(1 - n_0(\vec{x})) - z'_l(\vec{x})n_0(\vec{x})(1 - n_3(\vec{x})). \tag{35}$$

Moreover, in this step all the messenger bits are set to zero throughout the entire space. For any direction, the local absorption rule could be written more simply as

$$n'_a(\vec{x}) = n_a(\vec{x}) + z'_{-a}(\vec{x})z_a(\vec{x}) - z'_a(\vec{x})z_{-a}(\vec{x}). \tag{36}$$

Substituting in Eqs. (32) and (33) into Eq. (36), we have a single Boolean expression in terms of number variables for a single long-range interaction step for partition Γ_r as follows

$$n'_a(\vec{x}) = n_a(\vec{x}) +$$

Table 4: Long-range Interaction Sequence

events	$n_a(x)$	$z_l(x)$	$z_r(x)$	$n_a(x')$	$z_l(x')$	$z_r(x')$
initial	100000	0	0	000100	0	0
emit	100000	0	1	000100	1	0
kick	100000	1	0	000100	0	1
absorb	000100	0	0	100000	0	0

$$\begin{aligned} & n_a(\vec{x}+r\hat{e}_a)(1-n_{-a}(\vec{x}+r\hat{e}_a))n_{-a}(\vec{x})(1-n_a(\vec{x})) - \\ & n_{-a}(\vec{x}-r\hat{e}_a)(1-n_a(\vec{x}-r\hat{e}_a))n_a(\vec{x})(1-n_{-a}(\vec{x})) \end{aligned} \tag{37}$$

For convenience we define a long-range collision operator, P_a, as follows

$$P_a(\vec{x}) = z'_{-a}(\vec{x})z_a(\vec{x}), \tag{38}$$

so that we may write

$$n'_a(\vec{x}) = n_a(\vec{x}) + P_a(\vec{x}) - P_{-a}(\vec{x}). \tag{39}$$

The state data for this simple example we have been considering are given in Table 4, which represents all the steps of a long-range interaction sequence for a partition along the x-axis.

9 Another Example: Clockwise Orbit

To continue illustrating our implementation of a long-range lattice-gas, in this section we again consider a system with a single attractive interaction of range r, however the local momentum states participating in the interaction are not along the partition direction. However, in the example given here, the momentum exchange is still along the partition direction so that the long-range interaction remains a central-body one, resulting in a bound state with two particles trapped in a clockwise orbit. (Note that the restriction to central-body forces is not necessary, but is presented here for convenience.) In this slightly more complicated example, the local rules for photon emission and absorption, Eqs. (32) and (36) respectively, have a more general form with the implication that the emission and absorption of photons is different from the previous example of the bounce-back orbit, and the difference should be noted when making look-up tables to do this computation. The local photon emission rules can be written

$$\begin{aligned} z_a(\vec{x}) &= n_c(\vec{x})(1 - n_d(\vec{x})) \quad (40) \\ z_{-a}(\vec{x}) &= n_g(\vec{x})(1 - n_h(\vec{x})) \quad (41) \end{aligned}$$

where the bits c, d, g, and h must by chosen so momentum is conserved

$$\hat{e}_c - \hat{e}_d + \hat{e}_g - \hat{e}_h = 0 \quad (42)$$

as well as be constrained by central-body parallel and perpendicular momentum exchange conditions

$$\begin{aligned} (\hat{e}_c - \hat{e}_d - \hat{e}_g + \hat{e}_h) \cdot \vec{r} &= 2\Delta p \quad (43) \\ (\hat{e}_c - \hat{e}_d - \hat{e}_g + \hat{e}_h) \times \vec{r} &= 0, \quad (44) \end{aligned}$$

where Δp is the momentum change per site due to the long-range interaction. Equations (40) and (41) differ for Eqs. (30) and (31) for the bounce-back orbit by allowing two photons to be emitted at a single site.

To be explicit, for the two-dimensional triangular lattice, we can satisfy Eqs. (42), (43), and (44) by choosing the indices c, d, g, h as follows:

$$c = a - 2 \tag{45}$$

$$d = a - 1 \tag{46}$$

$$g = -c \tag{47}$$

$$h = -d. \tag{48}$$

An example of this choice of indices is illustrated in Figure 5c. Then the emission rules, Eqs. (40) and (41), are simply

$$z_a(\vec{x}) = n_{a-2}(\vec{x})[1 - n_{a-1}(\vec{x})] \tag{49}$$

Since the kicking of the photons is the same in this example as in the previous one, Eq. (33) still holds

$$z'_a(\vec{x} + r\hat{e}_a) = z_a(\vec{x}).$$

By re-expressing Eq. (36) more generally, we can write a local absorption rule

$$n'_a(\vec{x}) = n_a(\vec{x}) + z'_{-(a+1)}(\vec{x})z_{a+1}(\vec{x}) - z'_{a-1}(\vec{x})z_{-(a-1)}(\vec{x}) \tag{50}$$

or more elegantly

$$n'_a(\vec{x}) = n_a(\vec{x}) + P_{a+1}(\vec{x}) - P_{-a+1}(\vec{x}). \tag{51}$$

Table 5: Long-Range Interaction Sequence with Two Photons Emitted at a Single Site

events	$n_a(x)$	$z_l(x)$	$z_r(x)$	$n_a(x')$	$z_l(x')$	$z_r(x')$
initial	010010	0	0	000010	0	0
emit	010010	1	1	000010	1	0
kick	010010	1	0	000010	0	1
absorb	001010	0	0	000001	0	0

Substituting from Eqs. (49) and (33) into Eq. (50) and after some manipulation of the indices, we have a single boolean expression in terms of number variables for a single long-range interaction step for partition Γ_r as follows

$$\begin{aligned} n'_a(\vec{x}) &= n_a(\vec{x}) + \\ &\quad n_{a+2}(\vec{x}+r\hat{e}_{a+1})[1-n_{-a}(\vec{x}+r\hat{e}_{a+1}])n_{a-1}(\vec{x})[1-n_a(\vec{x})] - \\ &\quad n_{-a}(\vec{x}-r\hat{e}_{a-1})[1-n_{a-2}(\vec{x}-r\hat{e}_{a-1})]n_a(\vec{x})[1-n_{a+1}(\vec{x})]. \end{aligned}$$

Table 5 gives the local site data for the x-axis partition of a clockwise orbit. The particle $n_4(\vec{x})$ acts as a kind of spectator is this example, illustrating that two photons can be emitted from a single site. It is also possible to have two photons absorbed at a single site. Let us consider a *back-to-back* interaction over three sites. Suppose there are particles at sites $\vec{x} = 0$, $\vec{x}' = r\hat{\mathbf{i}}$, and $\vec{x}'' = 2r\hat{\mathbf{i}}$. Table 6 gives the site data for these sites where there are two photons emitted and absorbed at $\vec{x}'$ in the middle location.

The minimal model, using only an attractive interaction, models a fluid with liquid and gas phases at zero temperature. Figure 6 shows the time

Table 6: Long-Range Interaction Sequence with Two Photons Emitted and Absorbed at Site $\mathbf{x}'$ in a Back-to-Back Interaction

events	$n_a(x)$	$z_l(x)$	$z_r(x)$	$n_a(x')$	$z_l(x')$	$z_r(x')$	$n_a(x'')$	$z_l(x'')$	$z_r(x'')$
initial	010000	0	0	010010	0	0	000010	0	0
emit	010000	0	1	010010	1	1	000010	1	0
kick	010000	1	0	010010	1	1	000010	0	1
absorb	001000	0	0	001001	0	0	000001	0	0

evolution of the phase separation process in this case at a density $d = 0.07$ and interaction range $r = 6l$ and illustrates the type of physical simulation that can be achieved with the simplest long-range lattice-gas algorithm.

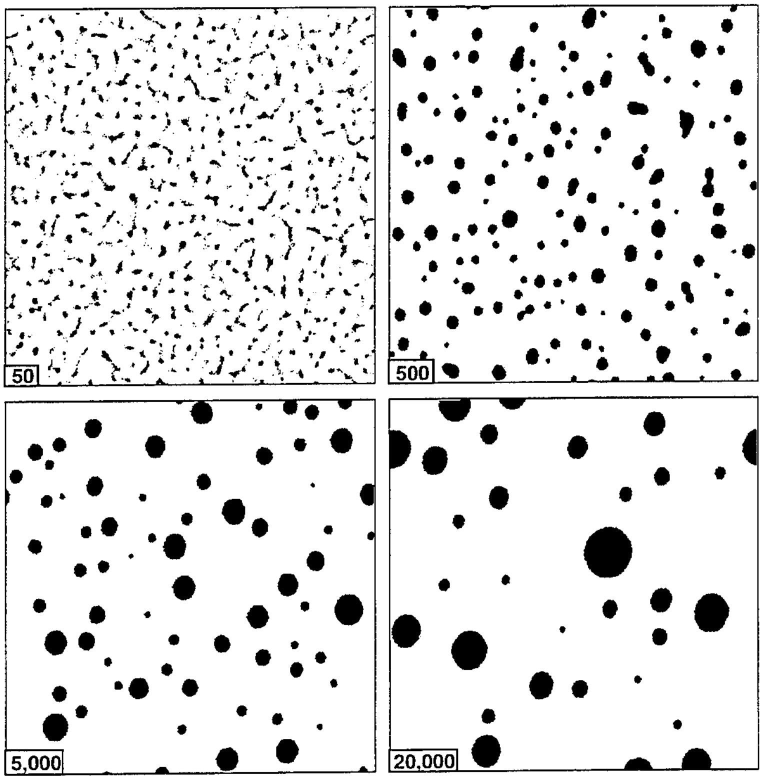

Figure 6: [Liquid-Gas Phase Separation. Time evolution of a liquid-gas phase separation for a lattice-gas with long range attractive interactions at range $r = 6$ on a 1024×1024 lattice starting with a uniformly random configuration of density $d = 0.07$.

10 Symmetries in Long Range Interactions

For an attractive interaction as mentioned earlier, we would expect that there exists a bound state in which two particles orbit one another, for example the clockwise orbit shown in Figure 5c. The range of the interaction is r. This hexagonal orbit is also shown in the bottom right corner of Figure 7. In this figure the hexagon's radius is also labeled as r and should not be confused with the interaction range, that is the hexagon's diameter. Simlarly a counter-clockwise orbit is possible. See the top left corner of Figure 7. A time-reversal invariance exists between these two cases with respect to conjugation and parity operations, as depicted in Figure 7.

These diagrams describe the possible 2-body collisions that can be so generated, including repulsive interactions. This logical correspondence between the different types of 2-body interactions allows one to achieve a more efficient implementation of a long-range lattice-gas algorithm than what we have achieved on the CAM-8, since we have not used this form of logical economy. Furthermore, because there is a correspondence property for $r = 0$, the situation reduces to the 2-body collisions in the FHP lattice-gas [2].

The long range interactions considered here have the following properties that simplify a computational implementation: 1) there exists only parallel momentum exchange, a restriction for modelling central body forces; 2) the

[2] For $r = 0$, the $|\Delta p| = 1$ interactions reduce to a rotation of the states, $\mathcal{R}(\frac{\pi}{3})$ and $\mathcal{R}(\frac{2\pi}{3})$; and the $|\Delta p| = 2$ interactions reduce to the identity operation.

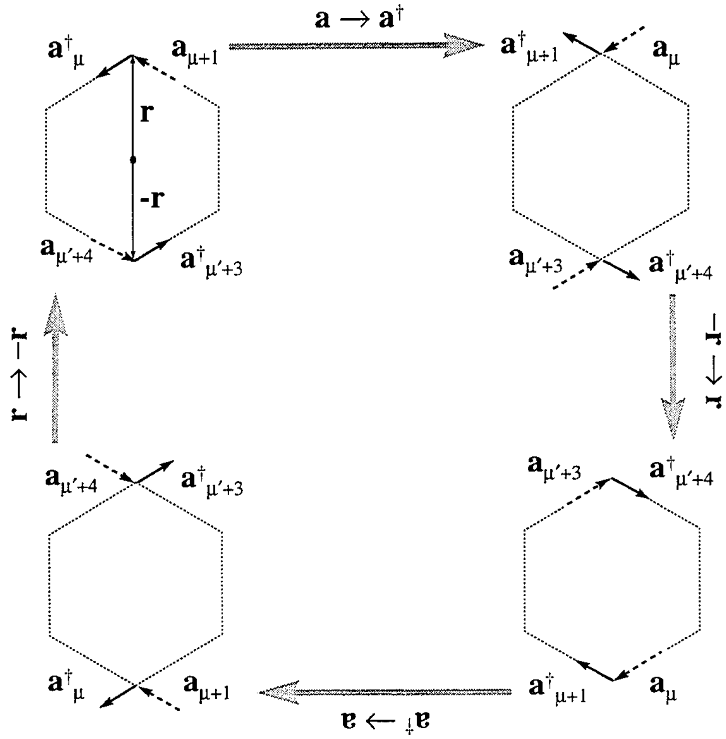
$\mathbf{a} \rightarrow \mathbf{a}^\dagger$
$\mathbf{a}^\dagger_\mu$
$\mathbf{a}_{\mu+1}$
$\mathbf{r}$
$\mathbf{-r}$
$\mathbf{a}_{\mu'+4}$
$\mathbf{a}^\dagger_{\mu'+3}$
$\mathbf{a}^\dagger_{\mu+1}$
$\mathbf{a}_\mu$
$\mathbf{a}_{\mu'+3}$
$\mathbf{a}^\dagger_{\mu'+4}$
$\mathbf{r} \rightarrow \mathbf{-r}$
$\mathbf{r} \rightarrow \mathbf{-r}$
$\mathbf{a}_{\mu'+4}$
$\mathbf{a}^\dagger_{\mu'+3}$
$\mathbf{a}_{\mu'+3}$
$\mathbf{a}^\dagger_{\mu'+4}$
$\mathbf{a}^\dagger_\mu$
$\mathbf{a}_{\mu+1}$
$\mathbf{a}^\dagger_{\mu+1}$
$\mathbf{a}_\mu$
$\mathbf{a}^\dagger \rightarrow \mathbf{a}$

Figure 7: Long-Range 2-Body Interaction Terms. Examples of two-body finite impact parameter collisions along the $\mathbf{r}_0$-direction. The four terms of the interaction Hamiltonian are for $|\Delta p| = 1$. Input states are depicted by dashed arrows and output states are by black arrows.

interaction acts only along the principal lattice directions; 3) there exists time-reversal invariance with respect conjugation and parity operations; and 4) a bias to the interactions can be assigned by coupling to a heat-bath reservoir. The previous examples, bounce-back and clockwise orbits, has been discussed properties 1 and 2. Property 3 has been discussed in this section. Property 4 has been discussed elsewhere [9].

11 Central-Body Interaction Neighborhood and 2D Crystallization

In the previous two sections, two examples of the long-range interactions, the bounce-back and clockwise orbits on a two dimensional triangular lattice were illustrated . In general, the long-range interaction step will involve many partitions, both attractive and repulsive interactions, and multiple ranges. In my CAM-8 implementation of a long-range lattice-gas with central-body interactions, I use 12 neighbors in two dimensions, as indicated in Figure 8. The triangular lattice is superposed over a square lattice, which appears rhomboidal in the figure. The square lattice is often used for embedding the site data into computer memory, which is rectilinear. This kind of embedding is the simplest and used for simulations that possess periodic boundary conditions. The reason for using 12 neighbors is to try to achieve a higher degree of local symmetry. In doing molecular dynamics modeling with a multi-long-range lattice-gas, we have found that 12 neighbors are necessary

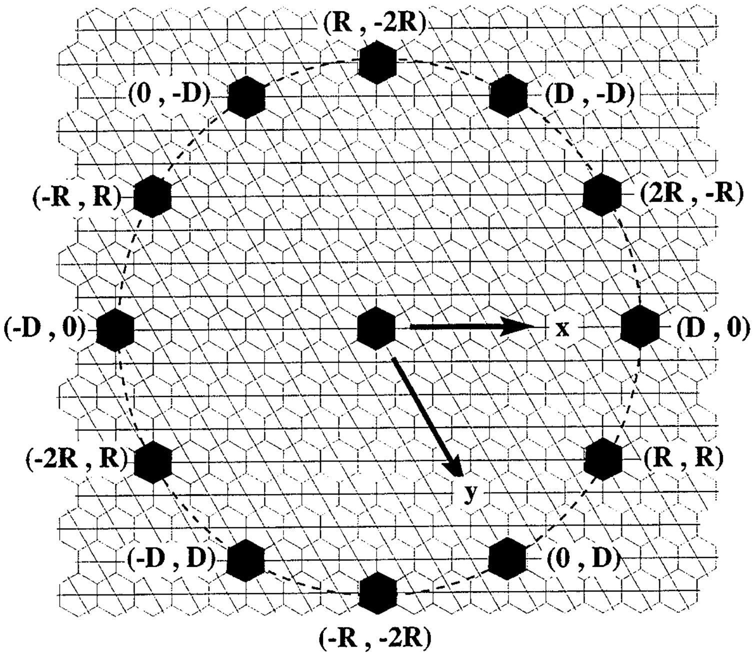

Figure 8: Twelve Neighbors on a Triangular Lattice that can Participate in Long-Range Central-Body Interactions. Interaction range $D = 7$ is depicted, where $R = \frac{D}{\sqrt{3}} \simeq 4$ to within 1.03 percent error. Computer memory space coordinates (x, y) are given adjacent to each neighboring site.

to recover macroscopic isotropy. In particular, 12 neighbors are necessary to have the emergent crystalline solid be able to freely rotate in space. A mean-field analysis of the lattice-gas crystallization method has been presented elsewhere [10]. Figure 8 shows a *ring* of range 7 lattice spacings.

To implement the crystallization algorithm, we use up to eight ranges in two dimensions, that is, eight rings of the type shown in Figure 8 for a total of 96 neighbors. Half the rings are used for attractive interactions

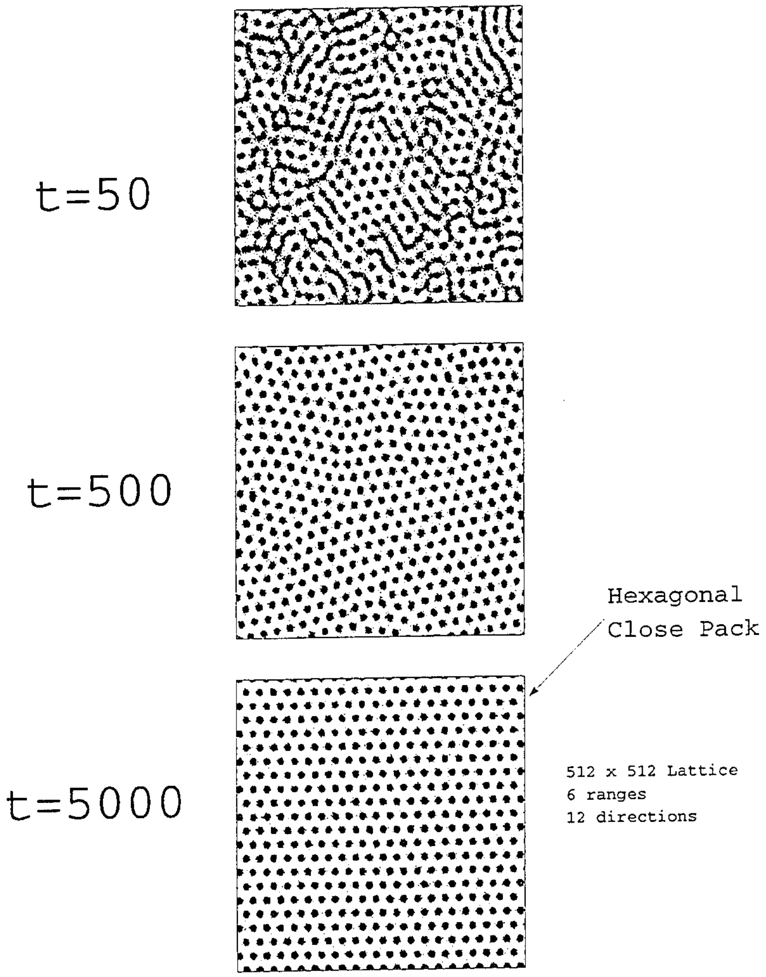

Figure 9: Time Evolution of Crystallization in a Two Dimensional Lattice-Gas with Multiple Fixed-Range 2-body Interactions. The resulting crystal is in a hexagonal-close-pack configuration since the coarse-grained interatomic potential is radially symmetric. The underlying lattice is 512 × 512. Started with a uniformly random configuration at $d = 0.1$. Twelve directions are used for long-range momentum exchanges. Grain boundaries and defects are observed during the early stages of the crystal formation.

and the other half are used for repulsive interactions. Typically, the inner rings are attractive, the middle rings are repulsive, and the outer rings are again attractive. Since four photon bits are used in our implementation, and since each ring is either attractive or repulsive, two rings are affected by a simultaneous partitioning of the space. For the attractive interaction, there are five types of orbits: bounce-back with $\Delta p = 2$, clockwise and counter-clockwise with $\Delta p = 1$, and clockwise and counter-clockwise with $\Delta p = \sqrt{3}$. Consequently, there are five types of repulsive interactions, which are just the conjugates of the five attractive ones. Since there are three partition directions for a triangular lattice, it takes 5*3=15 partitions of the space to completely update an attractive ring and a repulsive ring simultaneously. To compute 8 rings therefore takes 4*15=60 scans of the space. Therefore, since the local collisions require a single scan, it takes a total of 61 scans to complete one time step.

A two dimensional example using six interaction ranges, with an underlying 512 × 512 lattice, of this time-dependent crystallization process is given in Figure 9 and illustrates the type of molecular dynamics simulation that can be achieved with a more complex long-range lattice-gas algorithm. The resulting crystal is in a hexagonal-close-pack configuration since we have strived to make the coarse-grained interatomic potential be radially symmetric. This long-range lattice-gas model had six interaction ranges: $r = -2$,-7, 19, 21, -24, -26. Here the negative sign preceding the range denotes an attractive interaction at that range.

12 Conclusion

It should be noted that although the lattice-gas molecular dynamics algorithm that we have described above requires 61 scans, which is quite a lot of scans, this implementation only requires 10 bits of local site data (6 bits for the momentum states and 4 bits for messenger states) which means that only 1 kilobyte is needed to store a long-range rule. Since the CAM-8 uses a 16-bit word, there still remains 6 bits of unused local data. We use these remaining bits to hold a table look-up *address*. That is, since the size of the CAM-8 look-up table static random access memory (SRAM) is 64k bytes, and our long-range rule only requires 1k bytes, we can store up to 64 long-range rules into CAM-8's SRAM memory. Since our molecular dynamics algorithm decomposes into 61 applications of the long-range rules, it is now clear why we have chosen to use up to eight ranges. Although the description of our implementation may sound complicated, in fact from a software development point-of-view it was the most direct and most simple. We have traded off time to save memory. Yet the well known principle of computer science that one can save much time at the expense of using more memory applies to our algorithm. So optimizations of our algorithm can be made, particularly concerning trading off an increase of local site data for a decrease in the number of needed scans. Clearly, the molecular dynamics algorithms would be significantly sped up if thet were implemented say on a 64-bit architecture. Of course in this case, computation by table look-up would be inappropriate. However by making use of lattice isometries and rule conjugation, the necessary logic is actually quite small, as evidenced for example by Eq. (37) or (52). Therefore, a programmable logic method of computation should be better than the table look-up method of computation currently in use in the CAM-8 for this kind of lattice-gas algorithm.

References

[1] Frisch, U., Hasslacher, B., and Pomeau, Y (1986) Lattice-gas automata for the navier-stokes equation. *Phys. Rev. Lett.*, **56**(14):1505–1508.

[2] Margolus, N. and Toffoli T. (1990) Cellular automata machines. In Gary D. Doolean, editor, *Lattice Gas Methods for Partial Differential Equations*, pages 219–249. Santa Fe Institute, Addison-Wesley Publishing Company. The first 8-module CAM-8 prototype was operational in the fall of 1992.

[3] Margolus, N. (1993) Cam-8: a computer architecture based on cellular automata. In Ray Kapral and Anna Lawniczak, editors, *Proceedings of the Pattern Formation and Lattice-Gas Automata NATO Advanced Research Workshop*. Fields Institute for Research in Mathematical Sciences, American Mathematical Society.

[4] Toffoli, T. (1984) Cam: A high-performance cellular-automaton machine. *Physica*, **10**D:195–204. A demonstration TM-gas experiment was part of the CAMForth software distribution.

[5] Toffoli, T. and Margolus, N. (1987) *Cellular Automata Machines.* MIT Press Series in Scientific Computation. The MIT Press.

[6] Margolus, N., Toffoli, T., and Vichniac, G. (1986). Cellular-automata supercomputers for fluid-dynamics modeling. *Physical Review Letters*, **56**(16):1694–1696.

[7] Wolfram, S. (1986) Cellular automaton fluids 1: Basic theory. *J. of Stat. Phys.*, **45**(3/4):471–526.

[8] Appert, C., d'Humières D., Pot, V. and Zaleski, S. (1994) Three-dimensional lattice gas with minimal interactions. In *Transport Theory and Statistical Physics*, **23** (1-3), pages 107–122. Proceedings of Euromech 287 - Discrete Models in Fluid Dynamics, New York, M. Dekker. Editor P. Nelson.

[9] Yepez, J. (1993) A lattice-gas with long-range interactions coupled to a heat bath. In Ray Kapral and Anna Lawniczak, editors, *Proceedings of the Pattern Formation and Lattice-Gas Automata NATO Advanced Research Workshop.* Fields Institute for Research in Mathematical Sciences and NATO Advanced Research Workshop, American Mathematical Society.

[10] Yepez, J. (1994) Lattice-gas crystallization. *J. Stat. Phys.*, **81**(1/2):255–294.

[11] Appert, C. and Zaleski, S. (1990) Lattice gas with a liquid-gas transition. *Phys. Rev. Lett.*, **64**:1–4.

A Computer Implementation

To clearly illustrate the method of long-range lattice-gases, we have provided a computer implementation of this method for the prototype lattice-gas machine, the CAM-8. Further information concerning the CAM-8 prototype has been provided by Margolus [3]. The language we use to implement the model is called "CAMForth" and was developed at the MIT Laboratory for Computer Science as a superset of the Forth language provided on SUN SPARC-stations. Local computation on the CAM-8 parallel computer is achieved using look-up tables stored in fast static ram chips. Since the CAM-8 data word is 16 bits wide, a lookup table contains $64k$ entries. These tables are usually created off-line and stored on disk as binary encoded files, where the address of an individual entry is the input word, and the value of that entry is the output word. We have provided some C language subroutines that we use to create the look-up tables for our implementation. On the eight-module CAM-8 prototype, a 512×512 two-dimensional simulation runs at almost 400 frames per second. The interface dynamics, that is, coalescence of drops, for the type of simple liquid-gas model illustrated here, are clearly observable in real-time on the CAM-8's frame-buffer display or within an X-window on the host workstation.

A.1 Implementing a Lattice-Gas in the CAMForth Language

The following is a simple example of a long-range lattice-gas implemented in the CAMForth language. This is an Appert-type minimal lattice-gas [11] with an interaction range of 8. It is implemented here with periodic boundary conditions in two dimensions on the triangular lattice. The example code given below uses a 512 by 512 lattice size. We use a rhomboidal mapping of the triangular lattice onto the square lattice by using lattice directions: $\pm\hat{x}$, $\pm\hat{y}$, and $\pm(\hat{x}+\hat{y})$. Therefore, on the CAM-8's frame-buffer display, the representation of the lattice-gas on screen is sheared. Although it is possible to avoid this, for simplicity of coding, I have presented this example which is the most direct memory mapping. Our implementation uses a single messenger bit, or "photon", for each particle momentum state. Therefore, our implementation uses a total of 12 bits of local site data: 6 particle momentum states plus 6 photon states. Since we use six photon states the implementation actually does simultaneous updating of all lattice partitions.

All the dynamics are encoded into two 16-bit look-up tables. After the particle bits are kicked, the first look-up table, *lrlg.emit.lut*, actually performs two functions: (1) it does the local collisions; and (2) it then emits photons that are to be used in the long-range interaction step. After the photon bits are kicked, the second look-up table, *lrlg.absorb.lut*, performs local momentum changes by using the kicked photons, and in this way locally computes the long-range interaction. Any interaction range can be chosen, and in our code given below is set in the constant r. If a negative interaction range is set, the model still runs, however the interaction will be repulsive instead of being attractive. Therefore, a kind of parity operation with respect to the interaction range causes a conjugation of the interaction operation that results in flipping the polarity of the interparticle force.

```
\ ************* INITIALIZATION, SITE DATA, AND DATA MOVEMENT
new-experiment  512 by 512 space

0  0 == p0  1  1 == p1    2  2 == p2   3  3 == p3  4  4 == p4     5  5 == p5
6  6 == z0  7  7 == z1    8  8 == z2   9  9 == z3 10 10 == z4    11 11 == z5

: p-kicks kick   p0 field -1 x  0 y
                 p1 field  0 x -1 y
                 p2 field  1 x -1 y
                 p3 field  1 x  0 y
                 p4 field  0 x  1 y
                 p5 field -1 x  1 y
;
8  constant   r
r  negate constant  -r

: z-kicks kick   z0 field -r x  0 y
                 z1 field  0 x -r y
                 z2 field  r x -r y
                 z3 field  r x  0 y
                 z4 field  0 x  r y
                 z5 field -r x  r y
;
\ ************* PREPARE LOOK-UP TABLES AND DEFINE STEP
create-lut   collision.tab "" lrlg.emit.lut collision.tab load-buffer
create-lut interaction.tab "" lrlg.absorb.lut interaction.tab load-buffer

: prepare-tables
    lut-data   collision.tab switch-luts
    lut-data interaction.tab switch-luts
    step
; this is when-starting

define-step stepx
    site-src lut  lut-src site
    p-kicks run new-table z-kicks run new-table
end-step  this is update-step

\ ************* COLORMAP, DISPLAY TABLE, AND INITIAL PATTERN
"" lrlg.pal palette load-buffer palette>display
"" lrlg.dtab display-table load-buffer
"" lrlg.pat file>cam xvds
show-function xvds
```

This CAMForth example code, *lglr.fth*, loads several additional binary encoded files (to set up for frame-buffer visualization, data interaction through look-up table, and initial conditions) to run on the CAM-8 prototype or the CAM-8 simulator. These additional files are the following:

- *lrlg.pal* - Color palette used for display
- *lrlg.dtab* - Look-up table to render the density field
- *lrlg.pat* - Initial random pattern at 30
- *lrlg.emit.lut* - Local collisions and photon emission
- *lrlg.absorb.lut* - Long-range interaction table

A.2 Creating the Look-up Tables in the C Language

I have written a short subroutine in C language code to generate the look-up tables for this simple example. We will use the following notation for the number variables. The particle momentum states are $n(a) \equiv n_a$ and $\_n(a) \equiv 1 - n_a$ and $np(a) \equiv n'_a$ and $\_np(a) \equiv 1 - n'_a$. Similarly for the photon states, $z(a) \equiv z_a$ and $\_z(a) \equiv 1 - z_a$ and $zp(a) \equiv z'_a$ and $\_zp(a) \equiv 1 - z'_a$. The number variables for the particles and photons are then coded as global data in the C language as follows:

```
#define BIT(x,y) ((y>>x)%2)
#define CAM_WORD unsigned short
#define B 6

char   bit [B] ,  bitp [B] ;
char  zbit [B] , zbitp [B] ;

#define  n(a)  bit  [(a+B)%B]
#define _n(a)  (1-n(a))

#define  np(a)  bitp [(a+B)%B]
#define _np(a)  (1-np(a))

#define  z(a)  zbit  [(a+B)%B]
#define _z(a)  (1-z(a))

#define  zp(a)  zbitp [(a+B)%B]
#define _zp(a)  (1-zp(a))
```

It is possible to express the long-range lattice-gas dynamics as a logical function. Furthermore, it is possible to write a C language subroutine as a direct implementation of this logical function that accepts a CAM input word and computes the correct CAM output word returned by the subroutine. The first look-up table, *lrlg.emit.lut*, is generated by calling the following C language routine for all possible input configurations:

```
CAM_WORD lrlg_emit_lut (CAM_WORD input)
{
  char      a , C ;
  CAM_WORD  output=0  ;

  /*  FHP local collisions with z-emission */

  /* Convert input CAM word to input boolean variables */
  for ( a=0 ; a<B ; a++)  n(a) = BIT(a, input) ;

  for ( a=0 ; a<B ; a++)
    {
      C =  n(a+2) *  n(a+5) * _n(a+0) * _n(a+1) * _n(a+3) * _n(a+4)
        -  n(a+0) *  n(a+3) * _n(a+1) * _n(a+2) * _n(a+4) * _n(a+5)
        +  n(a+1) *  n(a+3) *  n(a+5) * _n(a+0) * _n(a+2) * _n(a+4)
        - _n(a+1) * _n(a+3) * _n(a+5) *  n(a+0) *  n(a+2) *  n(a+4) ;

      np(a) = n(a) + C ;
    }

  /* z-emission */
  for ( a=0 ; a<B ; a++)  z(a) = np(a+B/2) * _np(a) ;

  /* Convert output boolean variables to output CAM word  */
  for ( a=0 ; a<  B ; a++) output += np(a)<<a ;
  for ( a=B ; a<2*B ; a++) output +=  z(a)<<a ;

  return output ;
}
```

The low six bits hold the particle states, bits 0 to 5, and the high six bits hold the photon states, bits 6 to 11. The local collision operator contains the FHP 2-body and 3-body collisions [1]. The second look-up table, *lrlg.emit.lut*, is generated by calling the following C language routine for all possible input configurations

```
CAM_WORD lrlg_absorb_lut (CAM_WORD input)
{
  int      a ;
  CAM_WORD output=0   ;
  char       C ;


  /* Convert input CAM word to input boolean variables */
  for ( a=0 ; a<B ; a++) np(a) = BIT(a, input) ;

  /* Condition for outgoing photons */
  for ( a=0 ; a<B ; a++) z(a) = np(a+B/2) * _np(a) ;

  /* Streamed incoming photons */
  for ( a=B ; a<2*B ; a++)  zp(a) = BIT(a, input) ;

  /* Nonlocal interactions, photon absorption */
  for ( a=0 ; a<B ; a++)
    {
      C =  zp(a+B/2) * z(a)   -  zp(a) * z(a+B/2) ;
      n(a) = np(a) + C ;
    }

  /* Convert output boolean variables to output CAM word  */
  for ( a=0 ; a<B ; a++) output += n(a)<<a ;

  return output ;
}
```

We have provided the algorithm detail necessary for one to understand the essential computational structure of a long-range lattice-gas model. The example presented here of a liquid-gas system was coded in the CAMForth language and the C language. Code to generate display tables and color palettes was not presented here since that is not essential for understanding our lattice-gas implementation and is CAM-8 specific.